

handle

August 17, 2026

handle

Kind: Function**Source:** `src/adapter/service-worker/handler.ts`**Part of:** [Adapter](#)

Adapter for Service Worker

`handle` adapts Service Worker fetch events to the application's request handler. It receives an incoming request, runs it through the adapter flow, and returns the response used by the Service Worker.

Signature

```
function handle(app: Hono<E, S, BasePath>, opts: HandleOptions): Handler
```

Parameters

Name	Type
<code>app</code>	<code>Hono<E, S, BasePath></code>
<code>opts</code>	<code>HandleOptions</code>

Returns: `Handler`

Diagram

```
graph LR
  FetchEvent[Service Worker fetch event] --> Request[Request]
  Request --> Handle[handle]
  Handle --> App[Application handler]
```

```
App --> Response[Response]
Response --> FetchEvent
```

Usage

```
import { handle } from './handler'

self.addEventListener('fetch', (event) => {
  event.respondWith(handle(event.request))
})
```

AI Coding Instructions

- Keep the Service Worker adapter focused on converting incoming requests into application handler calls.
- Return the handler response directly so `fetch` events receive a valid `Response`.
- Register `handle` through `event.respondWith(...)` when wiring Service Worker fetch events.
- Do not add browser window APIs to this module; Service Worker code runs in a worker context.